Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method of operating a multithreaded parallel processor comprising: ~~computer instruction comprises:~~

directing the processor having a plurality of microengines to swap ~~a context swap instruction that swaps~~ a currently running context in a specified microengine out to memory to let another context execute in that microengine and ~~causes~~ cause a different context and associated program counter to be selected.

2. (Currently amended) The method ~~instruction~~ of claim 1 wherein the directing the processor ~~context swap instruction~~ wakes up the swapped out context when a specified signal is activated.

3. (Currently amended)The method ~~instruction~~ of claim 2 ~~1~~ wherein the signal is specified as a parameter in the directing ~~instruction~~ and specifies an occurrence of an event.

4. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "sram Swap", and the directing ~~context swap instruction~~ swaps out ~~the~~ a current context and wakes it up when the thread's SRAM signal is received.

5. (Currently Amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "sdram Swap," ~~"sram Swap",~~ the directing ~~context swap instruction~~ will swap out ~~the~~ a current context and wakes it up when the thread's SDRAM signal is received.

6. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies ~~as~~ "FBI" will swap out ~~the~~ a current context and wakes it up when the thread's FBI signal is received indicating that an FBI CSR, Scratchpad, TFIFO, or RFIFO operation has completed.

7. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "seq_num1_change/seq_num2_change", which swaps out ~~the~~ a current context and wakes it up when ~~the~~ a value of the sequence number changes.

8. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "inter_thread" which swaps out ~~the~~ a current context and wakes it up when the thread's interthread signal is received.

9. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "voluntary" which will swap out ~~the~~ a current context if another thread is ready to run, and if the thread is swapped, the swapped thread is automatically re-enabled to run at some subsequent context arbitration point.

10. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "auto_push" which swaps out ~~the~~ a current context and wakes it up when SRAM transfer read register data has been automatically pushed by ~~the~~ a FBus interface.

11. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "start_receive" that swaps out ~~the~~ a current context and wakes it up when new data in ~~the~~ a receive FIFO is available for this thread to process.

12. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "kill" which prevents ~~the~~ a current context or thread from executing again until ~~the~~ an appropriate enable bit for the thread is set in ~~the~~ a CTX_ENABLES register.

13. (Currently amended) The method ~~instruction~~ of claim 3 wherein the parameter specifies "pci" which swaps out ~~the~~ a current context and wakes it up when ~~the~~ a PCI unit signals that a DMA transfer has been completed.

14. (Currently amended) The method ~~instruction~~ of claim 3 wherein directing further comprises: ~~further comprising:~~

an optional_token "defer one" which specifies that one instruction will be executed after this reference before the context is swapped.

15. (Currently amended) A method of operating a multithreaded parallel processor comprises:

evaluating a specified parameter to determine a state of an executing context process; and

performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter.

16. (Original) The method of claim 15 wherein performing swaps a currently running context in a specified microengine out to memory to let another context execute in that microengine.

17. (Original) The method of claim 15 wherein the parameter specifies an occurrence of an event.

18. (Currently amended) The method of claim 15 wherein the parameter specifies "sram Swap", and performing a swapping comprises swapping out ~~the~~ a current context and ~~wakes~~ waking it up when the thread's SRAM signal is received.

19. (Currently amended) The method of claim 15 wherein the parameter specifies "sram Swap", and performing a swapping comprises swapping ~~the~~ a current context and ~~wake~~ waking it up when the thread's SDRAM signal is received.

20. (Currently amended) The method of claim 15 wherein the parameter specifies "inter_thread" which swaps out ~~the~~ a current context and wakes it up when the thread's interthread signal is received.

21. (Original) The method of claim 15 further comprising:

an optional_token "defer one" which specifies that one instruction will be executed after this reference before the context is swapped.

22. (Currently amended) A parallel processor that can execute multiple contexts and that comprises:

a register stack;

a program counter for each executing context;

an arithmetic logic unit coupled to the register stack and a program control store that stores a context swap instruction that causes the processor to:

evaluate a specified parameter to determine a state of an executing context process; and

perform a swap operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter and which saves an old program counter value.

23. (Currently amended) The processor of claim 22 wherein the context swap instruction wakes up the swapped out context when a specified signal is activated.

24. (Original) A computer program product residing on a computer readable medium for causing a multithreaded parallel processor to perform a function comprises instructions causing the processor to:

evaluate a specified parameter to determine a state of an executing context process; and

perform a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter.

25. (Original) The product of claim 24 wherein the processor wakes up the swapped out context when a specified signal is activated.